PQ SHIELD

Lattice-based Threshold Blind Signatures

NIST MPTS Workshop 2026 – 01/2026

Sebastian Faller, **Guilhem Niot**, Michael Reichle

Signatures now serve complex functions

Standard Signatures

Authenticate messages: proves origin and integrity

Advanced Signatures

Blind Signatures
Signer approves a message without seeing it

Ring Signatures
Signer hides in a group: untraceable signature

Anonymous credentials
Prove attributes without revealing your identity

Aggregate Signatures
Merge multiple signatures into a single one

Blind Signatures

User

"I would like a signature on my document without the signer seeing it"

Signer

Recover the final signature
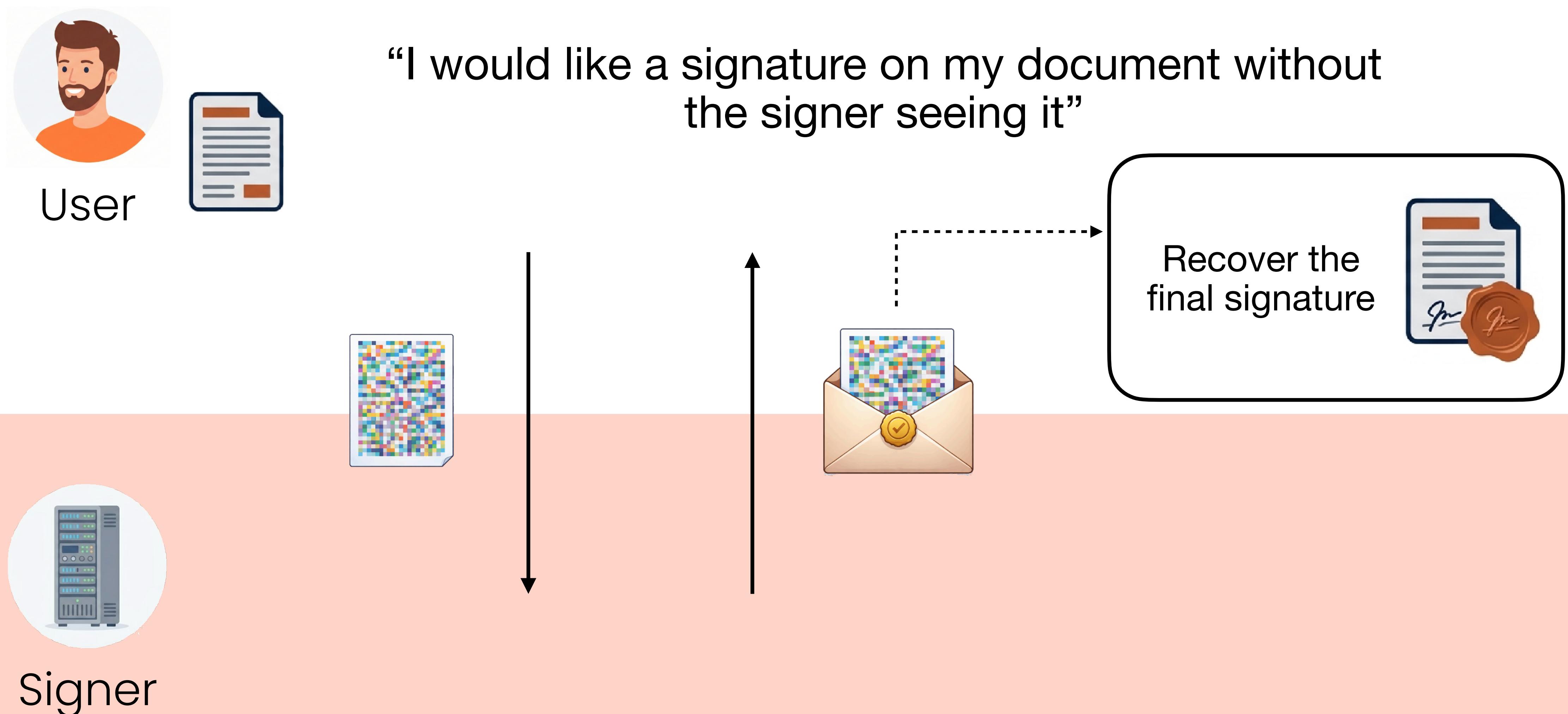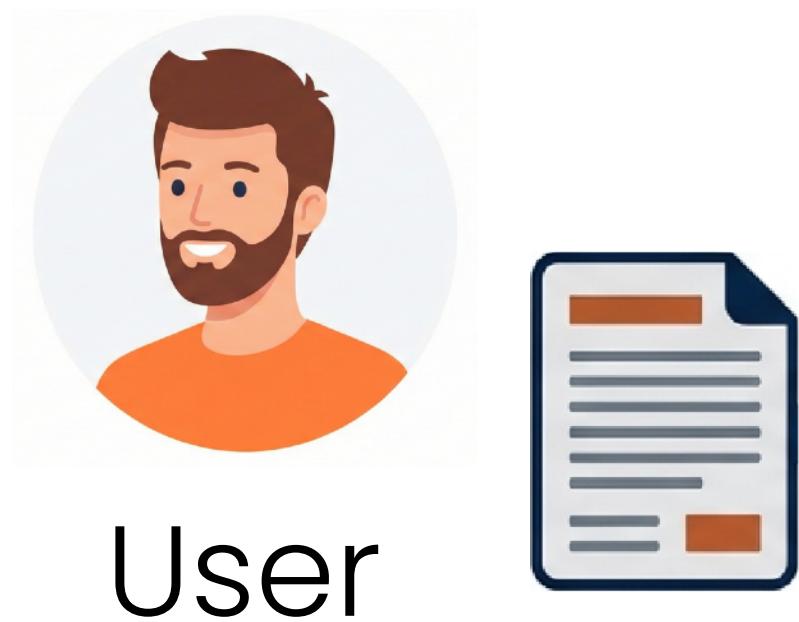
Blind Signatures

User

"I would like a signature on my document without the signer seeing it"

Applications

CBDCs

E-voting

Privacy Pass

Threshold Blind Signatures (TBS)

Blindness (User Privacy): Ensures the signer cannot link a signing request to the final signature or message content.

+
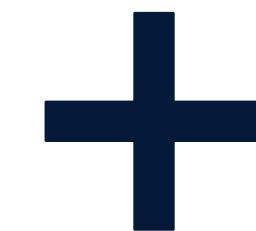
Threshold (Signer Security): Ensures no single party holds the full signing key; trust is distributed among N parties.

Threshold Blind Signatures (TBS)

Classical Setting

Post-Quantum Setting

Snowblind: A Threshold Blind Signature in Pairing-Free Groups

Elizabeth Crites[1], Chelsea Komlo[2], Mary Maller[3], Stefano Tessaro[4], and Chenzhi Zhu[4]

Stronger Security for Threshold Blind Signatures

Anja Lehmann[1], Phillip Nazarian[2], and Cavit Özbay[1]

Threshold Blind Signatures from CDH

Michael Reichle and Zoé Reinke

TBS has remained a significant open problem.

…

Question: How to achieve efficient TBS in the Post-Quantum setting?

Introducing the first Lattice-based TBS

Plover
(signature scheme)
[EENP+24]

One-More SIS
(Hardness assumption)
[AKSY22]

- Hash-and-sign
- Threshold-friendly

- Gives compact blind signatures
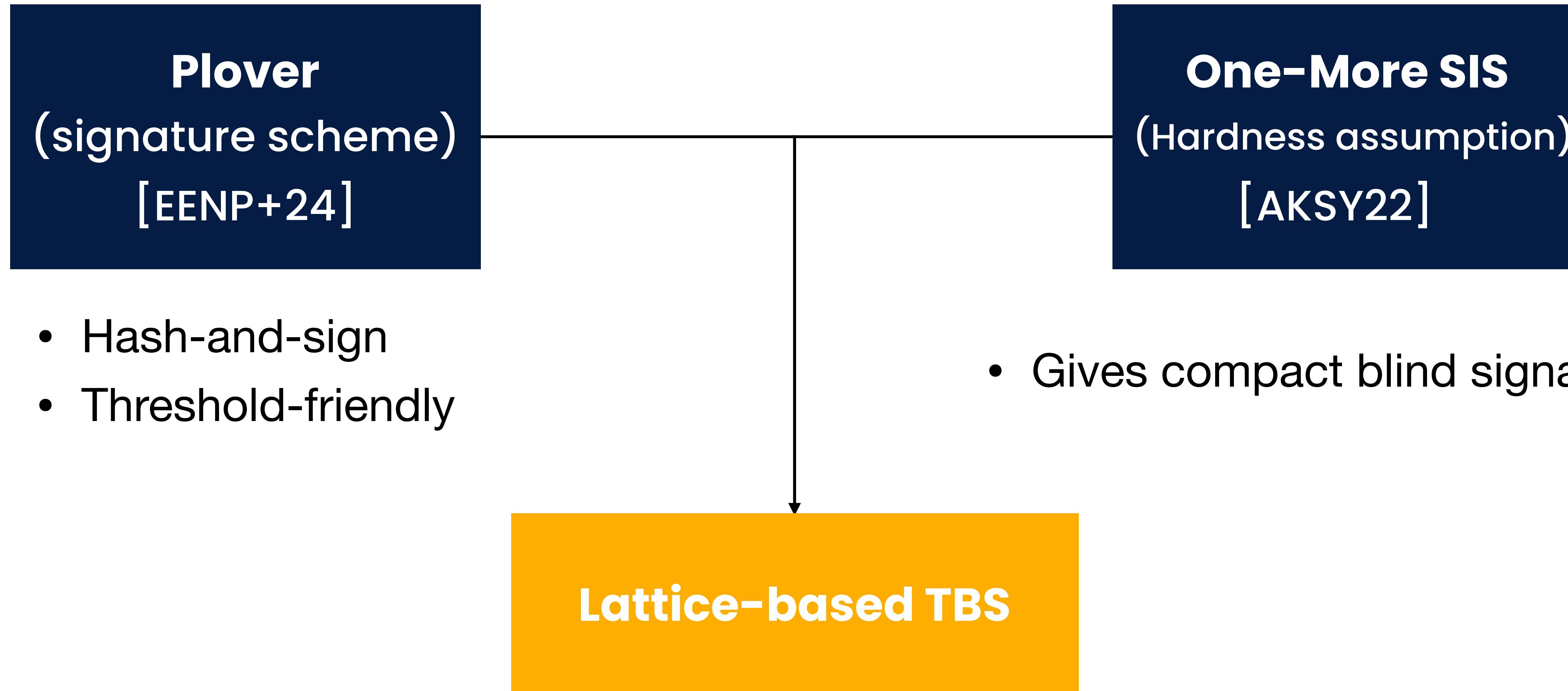
Lattice-based TBS

Introducing the first Lattice-based TBS

Plover

1. Derive $\text{target} = H(\text{msg}, \text{salt})$

2. Find sig such that $\mathbf{A} \cdot \text{sig} = \text{target}$

[AKSY22]

1. **User** derives $\text{target} = H(\text{msg}) + \mathbf{A} \cdot \text{noise}$

2. **Server** finds sig such that $\mathbf{A} \cdot \text{sig} = \text{target}$

3. **User** computes final signature $\text{sig} - \text{noise}$.

   Proves knowledge of it rather than give it in clear

... many technical challenges to achieve provable security.

Introducing the first Lattice-based TBS

Plover

[AKSY22]

Targets uniform vs adversarially chosen

1. Derive $\text{target} = H(\text{msg}, \text{salt})$

2. Find sig such that $\mathbf{A} \cdot \text{sig} = \text{target}$

1. **User** derives $\text{target} = H(\text{msg}) + \mathbf{A} \cdot \text{noise}$

2. **Server** finds sig such that $\mathbf{A} \cdot \text{sig} = \text{target}$

3. **User** computes final signature $\text{sig} - \text{noise}$.

   Proves knowledge of it rather than give it in clear

... many technical challenges to achieve provable security.

Introducing the first Lattice-based TBS

Plover

[AKSY22]

Original reduction for Gaussian signatures

1. Derive $\text{target} = H(\text{msg}, \text{salt})$

2. Find sig such that $\mathbf{A} \cdot \text{sig} = \text{target}$

1. **User** derives $\text{target} = H(\text{msg}) + \mathbf{A} \cdot \text{noise}$

2. **Server** finds sig such that $\mathbf{A} \cdot \boxed{\text{sig}} = \text{target}$

3. **User** computes final signature $\text{sig} - \text{noise}$.

   Proves knowledge of it rather than give it in clear

... many technical challenges to achieve provable security.

Introducing the first Lattice-based TBS

Plover

[AKSY22]

*Original reduction for **statistically** uniform matrix A*

1. Derive $\text{target} = H(\text{msg}, \text{salt})$

2. Find sig such that $\mathbf{A} \cdot \text{sig} = \text{target}$

1. **User** derives $\text{target} = H(\text{msg}) + \mathbf{A} \cdot \text{noise}$

2. **Server** finds sig such that $\boxed{\mathbf{A}} \cdot \text{sig} = \text{target}$

3. **User** computes final signature $\text{sig} - \text{noise}$.

   Proves knowledge of it rather than give it in clear

... many technical challenges to achieve provable security.

Introducing the first Lattice-based TBS

Set	Comm./party	Signature size
Compact *LaBRADOR NIZK*	46 KB	66 KB
Easy to deploy *LNP NIZK*	35 KB	115 KB

Easy to deploy set produces/verifies signatures in a few seconds.

Open problems

1. Explore applications

2. Hardness Assumptions

   Design efficient constructions from standard SIS or LWE assumptions.

3. Optimizations

   - Efficient compact set (LaBRADOR) implementation.
   - Leverage randomness reuse [JS24] to reduce signature size.

4. Advanced properties

   Support advanced properties such as robustness, identifiable abort, etc.

Questions?
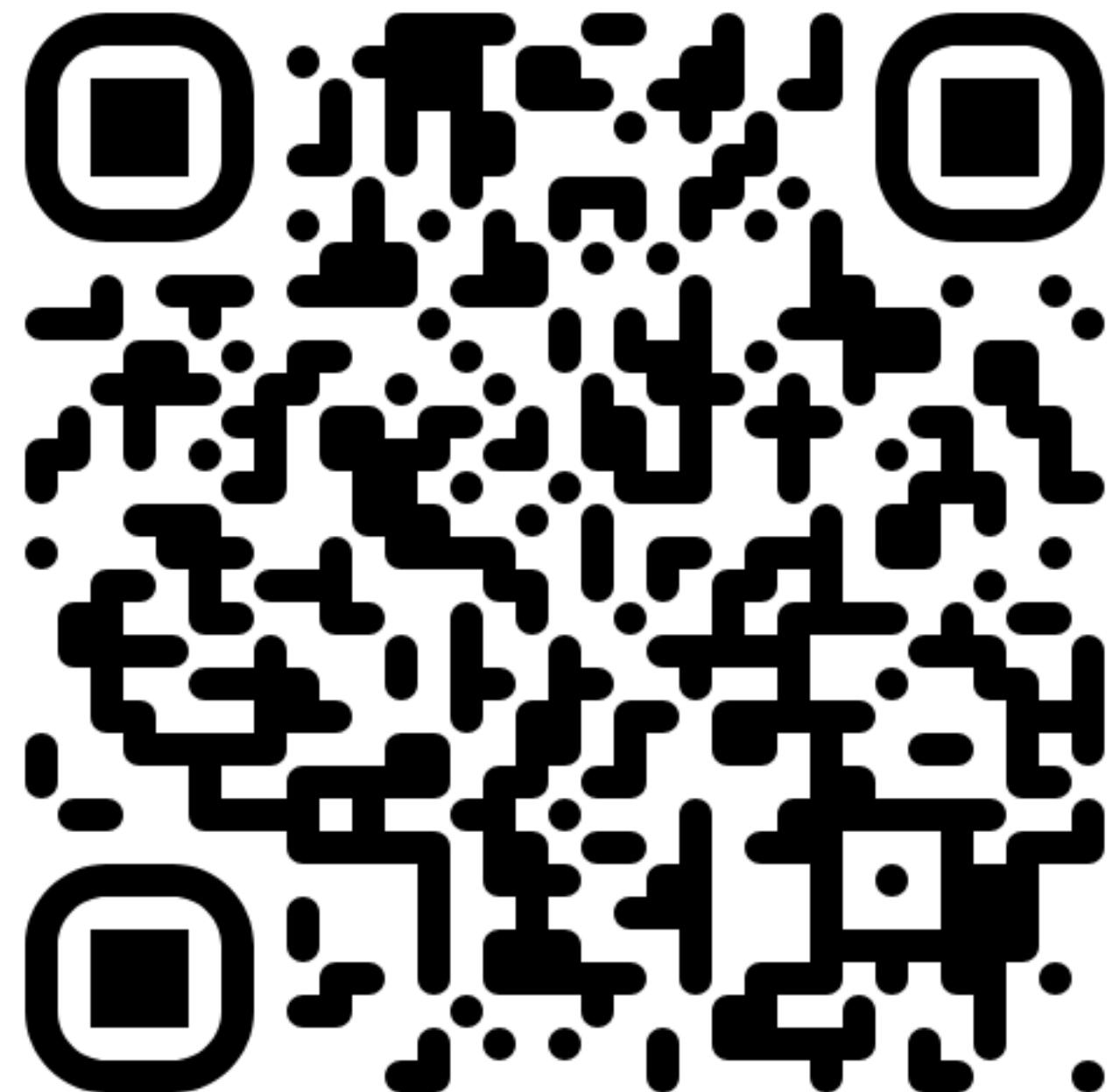
"**Lattice-based Threshold Blind Signatures**"
By Sebastian Faller, Guilhem Niot, Michael Reichle
Preprint
eprint.iacr.org/2025/1566